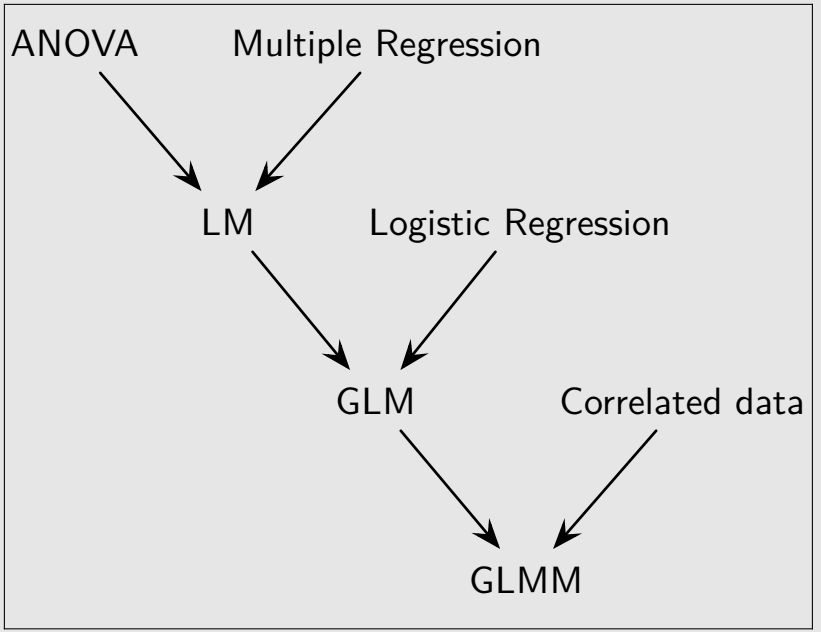
GLM tutoR

Outline

Highlights the connections between different class of widely used models in psychological and biomedical studies.



ANOVA vs. regression

Traditionally, ANOVA has been associated with the analysis of a continuous response and categorical predictors, or factors, in the tradition of design of experiments.
Consider the following model for a two-way ANOVA (factors A and B), without interaction:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ijk},$$

where y_{ijk} represents response of the kth subject for the ith level of A and jth level of B; μ is the overall mean, α_i and β_j are the main effects of A and B. (We need additional constraints to identify the model.)
If the design is balanced, the total variation in the responses can be partitioned in an orthogonal manner. An F-test can be constructed to test each of the two associated hypotheses (no effect of A and B).

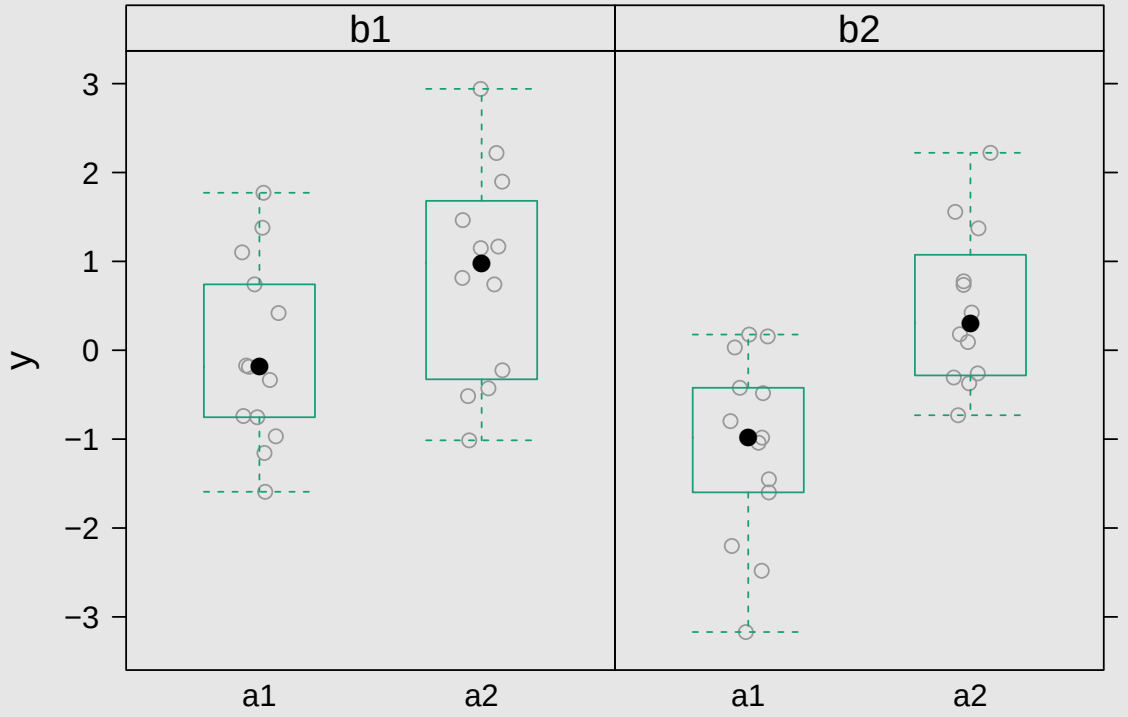
ANOVA vs. regression (Con't)

However, the standard ANOVA model can be formulated as a regression model where dummy-coded variables are used to code for the factors.
Let A and B be two-level factors, an equivalent regression model is

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i,$$

with β_0 is the intercept, β_1 and β_2 the regression coefficients associated to each factor.
An F-test can be construct to test whether any of the regression coefficients is different from zero, while t-tests can be used for testing each regression coefficient.

A simulated dataset



aov or lm?

R's aov function is just a wrapper function for lm, except that it adds a special treatment to the Error= term. More precisely, lm uses the residual error as the error term for all effects

Sample size
```
n <- 50
```
Effect sizes
```
es <- c(.6, -.4)
```
Two-level factors, A and B, with [-1,1] contrast
```
A <- gl(2, 2, n, labels=c(-1,1))
B <- gl(2, 1, n, labels=c(-1,1))
```
Simulate a response vector
```
y <- es[1]*as.numeric(as.character(A)) +
     es[2]*as.numeric(as.character(B)) +
     rnorm(n)
```
Output from ANOVA and LM
```
summary(aov.res <- aov(y ~ A + B))
summary(lm.res <- lm(y ~ A + B))
```

aov or lm? (Con't)

Results for `aov(y ~ A + B)` are:

```
            Df Sum Sq Mean Sq F value   Pr(>F)
A            1  18.88  18.884   16.90 0.000157 ***
B            1   6.69   6.695    5.99 0.018176 *
Residuals   47  52.53   1.118
```

while for `lm(y ~ A + B)` we have:

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.2016     0.2556  -0.789 0.434292
A1            1.2301     0.2993   4.111 0.000157 ***
B1           -0.7318     0.2990  -2.448 0.018176 *
```

The $\hat{\beta}_1$ coefficient (`A1`) is readily obtained as `diff(tapply(y, A, mean))`.

However, using `anova(lm(y ~ A + B))` yields the expected ANOVA table.

A model comparison approach

We can test the significance of a single predictor using an F-test:

Fit a reduced model	`lm.res2 <- update(lm.res, . ~ . - B)`
Compute reduction in RSS	`anova(lm.res2, lm.res)`

The overall F-value in the regression table is simply `anova(lm(y ~ 1), lm.res)` (with 2 df, for the difference in terms of parameters). In other words, we compare a model with k parameters to the base or null model which includes only an intercept term (the 'grand mean').

This works for ANOVA too: `anova(aov(y ~ A), aov(y ~ A+B))` yields the F-test corresponding to the main effect of B.

In this particular setting, we will get similar results using `drop1(lm.res, test="F")`. (With a balanced design, Type I, II, and III SS all give the same results.)

Types of sum of squares

Testing the effects of ANOVA terms with unbalanced data involves choosing the way SS are computed since the factors are no longer orthogonal, e.g. [1] and [2, §8.2.4–8.2.6].
The difference between Type I/II and Type III (also called Yates's weighted squares of means) lies in the model that serves as a reference model when computing SS, and whether factors are treated in the order they enter the model or not. E.g., for a saturated two-way ANOVA model:

- Type I (default): SS(A), SS($B|A$), then SS($AB|B$, A)
- Type II: SS($A|B$), then SS($B|A$) (no interaction)
- Type III: SS($A|B$, AB), SS($B|A$, AB) (interpret each main effect after having accounted for the other main effect and interaction)

Types of sum of squares (Con't)

Here is an illustration of how SSs will differ with an unbalanced design:

Load package	`library(car)`
Mirror our dataset but	`tmp <- data.frame(y, A, B)`
remove some combinations of factor	`tmp <- tmp[-sample(1:n, 10),]`
Fit a full model	`aov.res2 <- aov(y ~ A * B, data=tmp)`
Type II SS	`Anova(aov.res2)`
Type III SS	`Anova(aov.res2, type="III")`

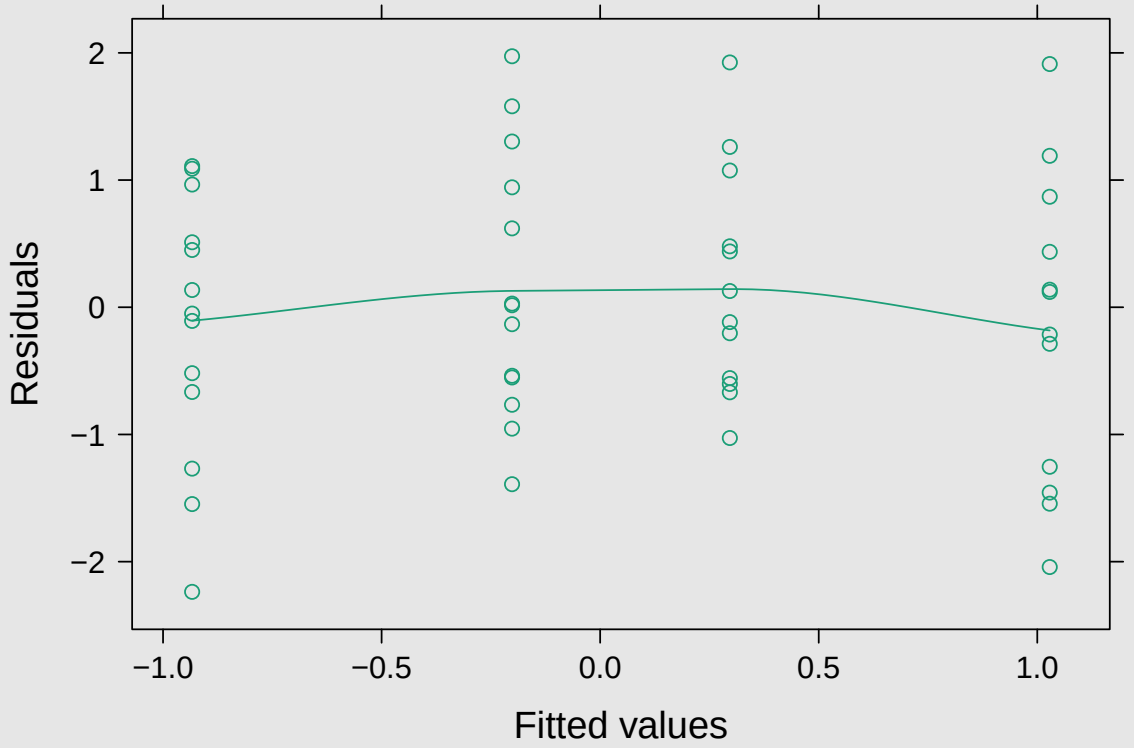
Model diagnostics

Don't trust your model without examining the quality of fit, that is the distribution of residuals!
Two useful diagnostic plots:

- Plot residuals versus predicted response to verify that variance is constant and that no outliers are present.
- Plot residuals against each predictor to verify the linearity of the modelled relationships. With multiple predictor, it is called a partial residual plot.

Partial residual (Ceres) plots are available as `car::crPlot` or `faraway::prplot`.

Model diagnostics (Con't)



Model predictions

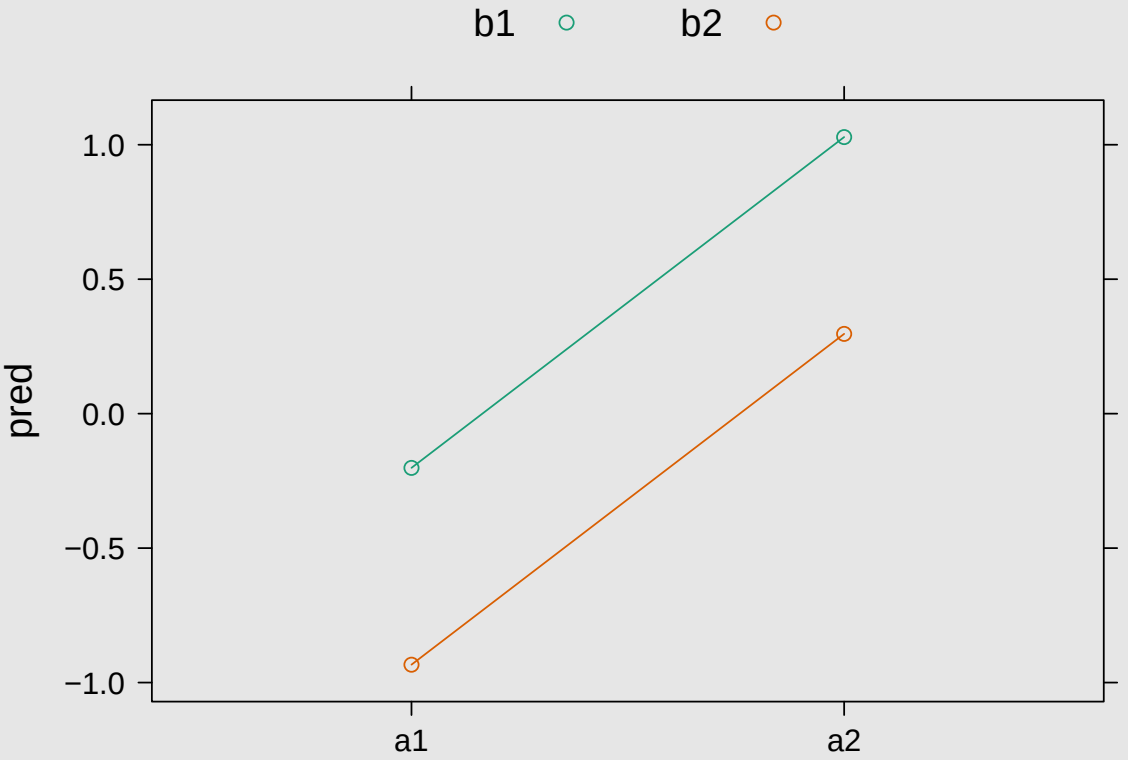
The ANOVA has four cells (and the rhs of the regression model yields four possible outcome for $\widetilde{y}_i$). In R, we can use

Combination of factor levels	`new.df <- expand.grid(A=levels(A), B=levels(B))`
Predict expected values at a glance	`new.df$pred <- predict(lm.res, new.df)`
Or using little algebra ($a_1 b_2$)	`t(coef(lm.res)) %*%c (1, 0, 1)`

Adding `se.fit=TRUE` will give the corresponding standard errors, while confidence intervals for predicting future observations are obtained with `interval="p"`.

```
   A  B        fit       lwr      upr
1 -1 -1 -0.2015823 -2.389632 1.986468
2  1 -1  1.0285051 -1.162855 3.219865
3 -1  1 -0.9334258 -3.121476 1.254624
4  1  1  0.2966616 -1.894699 2.488022
```

Predicted values



Computing CIs for model parameters

The `confint` function gives asymptotic confidence intervals for a certain level (default, 95%):

```
                2.5 %     97.5 %
(Intercept) -0.7158092  0.3126446
A1           0.6280653  1.8321095
B1          -1.3333837 -0.1303032
```

Instead of relying on asymptotic distribution, we could also estimate 95% CIs using bootstrap.

Load required package
```
library(boot)
```

The additive model
```
fm <- y ~ A + B
```

Create a placeholder
```
dd <- data.frame(y, A, B)
```

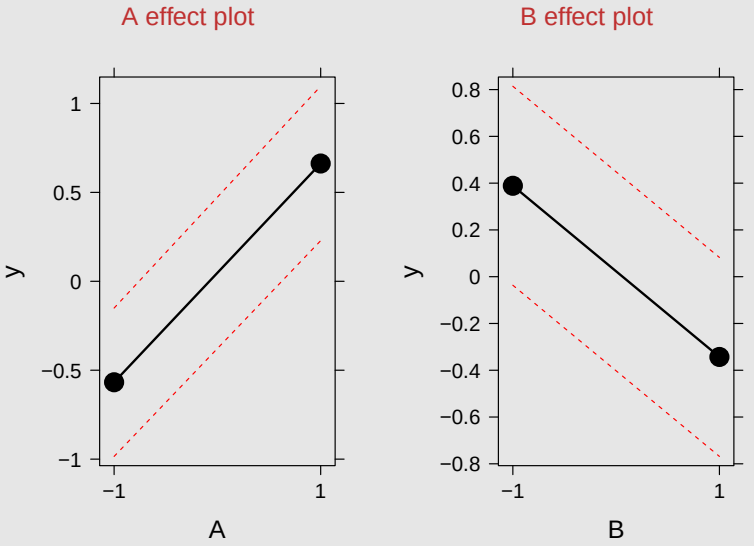
Function that returns model parameters
```
reg.boot <- function(formula, data, k)
  coef(lm(formula, data[k,]))
```

Main call to the bootstrap procedure
```
reg.res <- boot(data=dd, statistic=reg.boot,
                R=500, formula=fm)
```

Bias-corrected confidence intervals for A
```
boot.ci(reg.res, type="bca", index=2)
```

Studying effects

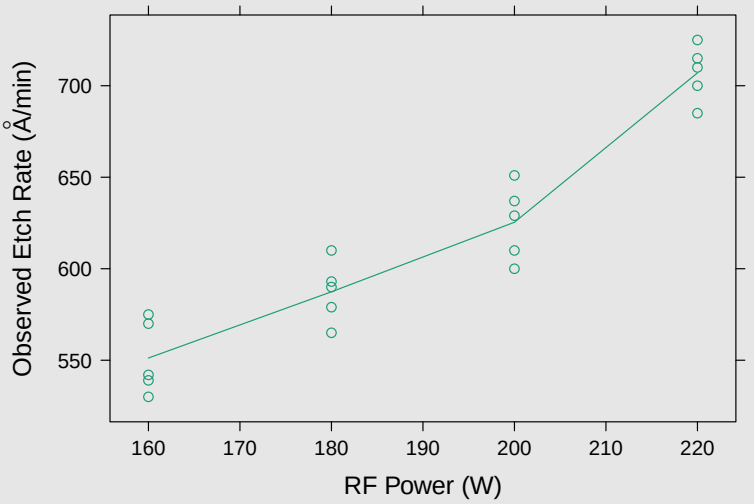
For ANOVA, various measures of effect size have been proposed in the literature. Some of them are available in the `MBESS` package.
For graphical displays, useful functions are available in the `effects` package [3].



Case study: The Etch Rate (ER) data

Data on etch rate as a function of RF Power [4].

Load the dataset	`etch.rate <- read.table("etchrate.txt", h=T)`
Display raw and average values	`xyplot(rate ~ RF, etch.rate, type=c("p","a"))`



Case study: The ER data (Con't)

The effect model reads

$$y_{ij} = \mu + \tau_i + \varepsilon_{ij}, \qquad i = 1, \ldots, a;\ j = 1, \ldots, n$$

where τ_i represent the difference between treatment means and the overall ('grand') mean, and $\varepsilon_{ij} \underset{\text{iid}}{\sim} \mathcal{N}(0; \sigma^2)$.

Fitting the one-way ANOVA in R is done as follows:

Convert each variable to factor

```
etch.rate$RF <- as.factor(etch.rate$RF)
etch.rate$run <- as.factor(etch.rate$run)
```

Fit the model

```
etch.rate.aov <- aov(rate~RF,etch.rate)
summary(etch.rate.aov)
```

The ANOVA table is given below:

```
            Df Sum Sq Mean Sq F value   Pr(>F)
RF           3  66871   22290    66.8 2.88e-09 ***
Residuals   16   5339     334
```

Case study: The ER data (Con't)

A $100(1-\alpha)\%$ confidence interval for treatment effect $\hat{\tau}_i = \bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot}$ (see `model.tables`) is computed as

$$\bar{y}_{i\cdot} \pm t_{\alpha/2,N-a}\sqrt{\frac{\text{MSE}}{n}},$$

whereas for any two treatment comparison the above formula becomes

$$(\bar{y}_{i\cdot} - \bar{y}_{j\cdot}) \pm t_{\alpha/2,N-a}\sqrt{\frac{\text{MSE}}{n}}.$$

We only need to compute the pooled SD which is returned by `summary`.

Extract MS error	`MSe <- summary(etch.rate.aov)[[1]][2,3]`
Compute pooled SD	`SD.pool <- sqrt(MSe/5)`
Critical quantile of Student's t	`t.crit <- c(-1,1) * qt(.975,16)`

Case study: The ER data (Con't)

Here, any two treatment difference has an associated 95% CI of $(\bar{y}_{i\cdot} - \bar{y}_{j\cdot}) \pm 24.5$, slightly narrower compared to a t-test.
Note, however, that `confint(etch.rate.aov)` will yield different results:

```
                2.5 %    97.5 %
(Intercept) 533.88153 568.51847
RF180        11.70798  60.69202
RF200        49.70798  98.69202
RF220       131.30798 180.29202
```

What is computed here is the estimated 95% CI for treatment effect substracted to a baseline (or reference level, 160 W) because R uses treatment contrast by default. We can compute the last row as

Group means

```
grp.means <- with(etch.rate, tapply(rate, RF,
                                    mean))
```

95% CI for (RF220-RF160)

```
as.numeric(grp.means[4]-grp.means[1]) +
  c(-1,1) * qt(.975,16) * sqrt(2*MSe/5)
```

Case study: The ER data (Con't)

More complex contrasts, especially with higher-order terms in ANOVA models, can be computed with the `multcomp` package.
For example, the preceding contrast would be obtained as follows:

Load package	`library(multcomp)`
Fit a linear model	`etch.rate.lm <- lm(rate ~ RF,etch.rate)`
Create the contrast of interest	`etch.rate.glht <- glht(etch.rate.lm, mcp(RF=c(-1,0,0,1)))`
Show p-value	`summary(etch.rate.glht)`
Display associated 95% CI	`confint(etch.rate.glht)`

What are GLMs?

The theory of Generalized Linear Model encompasses a unified approach to regression models where a single response variable is assumed to follow one of the exponential family probability distribution [5]. This includes the following PDFs: gaussian, binomial, Poisson, gamma, inverse Gaussian, geometric, and negative binomial.
The idea is to 'relaxe' some of the assumptions of the linear model such that the relationship between the response and the predictors remains linear. You may recall that in the case of linear regression, we usually relate the predictors to the expected value of the outcome like so:

$$\mathbb{E}(y \mid x) = \mathbf{X}\beta$$

From linear to logistic regression

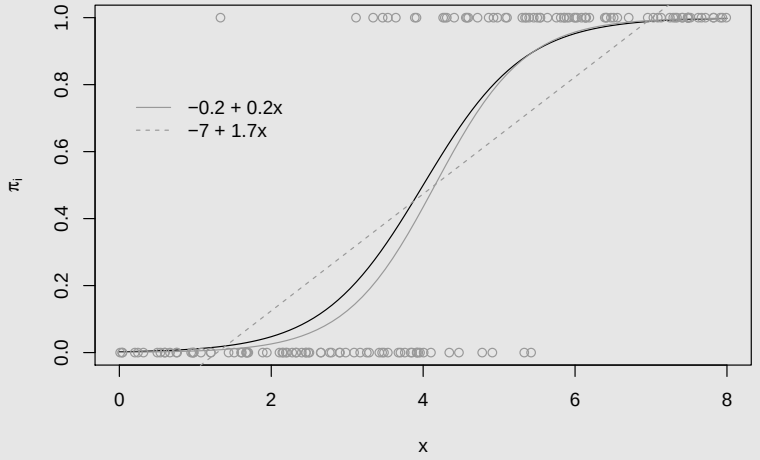
How can this be achieved with a logistic regression where individual responses are binary and follow a Bernoulli, or $\mathcal{B}(1; 0.5)$, distribution? Moreover, a standard regression model could predict individual probabilities outside the $[0; 1]$ interval.
Some transformations, like $p' = \arcsin p$, have been proposed to allow the use of ANOVA with binary data [6, p. 278–280]. However, it is fairly easy to apply a logistic regression, see also [7].

Considering the logit transformation of the probability of the event under consideration, $\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$, the logistic regression model is comparable to the linear case, i.e. it is additive in its effect terms. In the simplest case (one predictor + an intercept term), we have:

$$g(x) = \ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x.$$

Illustration with artificial data

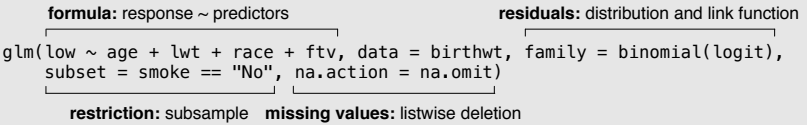
Suppose the following model holds: $\pi_i = \frac{\exp(-6+1.5x_i)}{1+\exp(-6+1.5x_i)}$, where π_i is the probability of a positive outcome for individual i. Below are the results from fitting a logistic regression and a linear regression on $N = 150$ observations drawn from the above model.



At a glance

Every commands that can be used with linear regression (`predict`, `fitted`, `resid`, `summary`, `plot`, etc.) will work when fitting a logistic model. However, there are more convenient functions in the `rms` package [8].

Instead of `lm`, we will now use `glm`:



The `effects` package also works with GLMs.

Case study: The lwb study

Prognostic study of risk factor associated with low birth infant weight [9].

Load the dataset

```
data(birthwt, package=MASS)
```

Recode categorical predictors

```
birthwt <- within(birthwt, {
  race <- factor(race, labels=c("White",
                                "Black",
                                "Other"))
  smoke <- factor(smoke, labels=c("No","Yes"))
  ui <- factor(ui, labels=c("No","Yes"))
  ht <- factor(ht, labels=c("No","Yes"))
})
```

Some descriptive statistics

```
library(Hmisc)
summary(low ~ age + lwt + race + ftv,
        data=birthwt)
```

Case study: The lwb study (Con't)

		N	low
age	[14,20)	51	0.2941176
	[20,24)	56	0.3571429
	[24,27)	36	0.4166667
	[27,45]	46	0.1956522
lwt	[80,112)	53	0.4716981
	[112,122)	43	0.2325581
	[122,141)	46	0.2608696
	[141,250]	47	0.2553191
race	White	96	0.2395833
	Black	26	0.4230769
	Other	67	0.3731343
ftv	0	100	0.3600000
	1	47	0.2340426
	2	30	0.2333333
	3	7	0.5714286
	4	4	0.2500000
	6	1	0.0000000
Overall		189	0.3121693

Case study: The lwb study (Con't)

We will now consider the following model, in Wilkinson and Rogers' notation: `low ~ age + lwt + race + ftv`.

Load package	`library(rms)`
Update the current environment	`ddist <- datadist(birthwt)` `options(datadist="ddist")`
Fit a logistic regression	`fit.glm1 <- lrm(low ~ age + lwt + race + ftv,` `data=birthwt)`
Display the results	`print(fit.glm1)`

Here, the default link function is a `binomial(logit)`. The above model is equivalent to `glm(low ~ age + lwt + race + ftv, data=birthwt, family=binomial)`.

Case study: The lwb study (Con't)

Below is a partial output showing estimates for the regression coefficients (on the link scale):

```
             Coef    S.E.   Wald Z Pr(>|Z|)
Intercept    1.2954 1.0714  1.21   0.2267
age         -0.0238 0.0337 -0.71   0.4800
lwt         -0.0142 0.0065 -2.18   0.0294
race=Black   1.0039 0.4979  2.02   0.0438
race=Other   0.4331 0.3622  1.20   0.2318
ftv         -0.0493 0.1672 -0.29   0.7681
```

In place of the overall F-test for a regression table, we now have a likelihood ratio test for the model. The principle is identical (assess the reduction in deviance between the null model and the full model).

```
Model Likelihood
  Ratio Test
LR chi2     12.10
d.f.            5
Pr(> chi2) 0.0335
```

Case study: The lwb study (Con't)

A more convenient summary table (adjusted ORs with 95% CI) might be obtained with `summary(fit.glm1)`.

```
            Effects                Response : low

 Factor              Low High Diff. Effect S.E. Lower 0.95 Upper 0.95
 age                  19   26    7  -0.17  0.24 -0.63       0.30
  Odds Ratio          19   26    7   0.85    NA  0.53       1.34
 lwt                 110  140   30  -0.43  0.20 -0.81      -0.04
  Odds Ratio         110  140   30   0.65    NA  0.44       0.96
 ftv                   0    1    1  -0.05  0.17 -0.38       0.28
  Odds Ratio           0    1    1   0.95    NA  0.69       1.32
 race - Black:White    1    2   NA   1.00  0.50  0.03       1.98
  Odds Ratio           1    2   NA   2.73    NA  1.03       7.24
 race - Other:White    1    3   NA   0.43  0.36 -0.28       1.14
  Odds Ratio           1    3   NA   1.54    NA  0.76       3.14
```

Case study: The lwb study (Con't)

Like in linear regression, to assess the significance of individual predictors we can compare two nested models. E.g., `age` and `ftv` appear to be non-significant. An LRT can confirm that:

Fit a reduced model	`fit.glm2 <- update(fit.glm1, . ~ . -age-ftv)`
Test it against the full model	`lrtest(fit.glm2, fit.glm1)`
Summarize the reduced model	`anova(fit.glm2)`

Stepwise methods for variable selection can also be used but be aware that such methods are unstable, yield biased estimation of regression coefficients and misspecified estimates of variability, but above all there is no control on p-values. See [8] or [10, §11.7] for alternative ways of performing variable selection.

Case study: The lwb study (Con't)

Using the last model, we can predict the expected outcome (with confidence intervals for means) for a particular range of weight at last menstrual period, depending on mother's ethnicity. The syntax is similar to that of R base `predict` function.

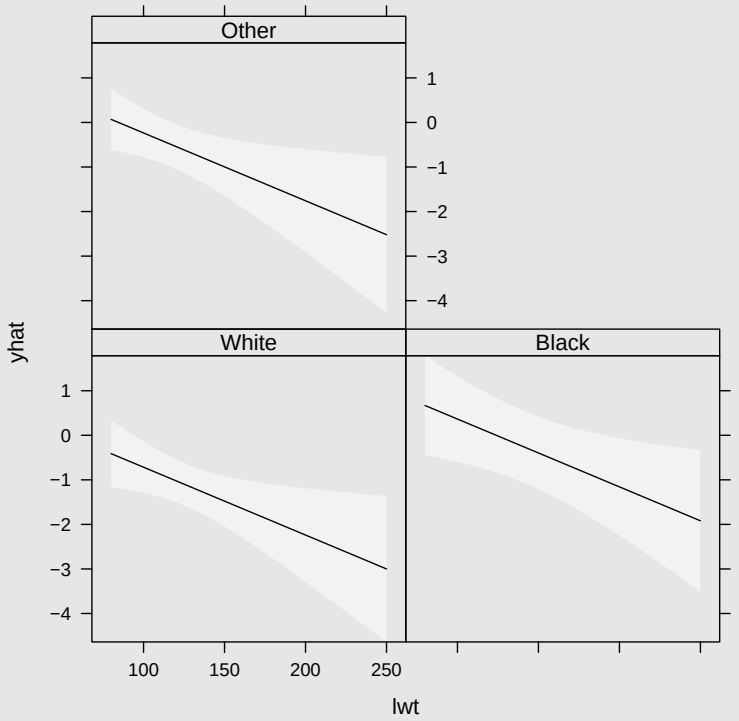
Still on the log odds scale, we can predict the expected weight category for a white mother's of last menstrual weight `lwt=150`.

Predicted response on the log odds scale	`pred.glm2 <- Predict(fit.glm2, lwt=seq(80, 250, by=10), race)`
Point estimate for a particular outcome	`Predict(fit.glm2, lwt=150, race="White")`
Get the OR using little algebra	`exp(sum(coef(fit.glm2)*c(1, 150, 0, 0)))`

```
  lwt  race      yhat     lower      upper
1 150 White -1.477712 -2.042931 -0.9124926

Response variable (y): log odds
```

Case study: The lwb study (Con't)



Take-away message

- ANOVA, Linear regression, and Logistic regression can be understood using the GLM framework which related a response variable to continuous or categorical predictors through a link function.
- R provides a lot of tools to construct GLMs, assess model fit, and display summary statistics either graphically or numerically.
- Rather than p'ing everything, it is often more interesting (and informative!) to work with interval and pointwise estimation, together with effect size.

References

[1] DG Herr. On the history of anova in unbalanced, factorial designs: The first 30 years. *The American Statistician*, 40(4):265–270, 1986.

[2] J Fox. *Applied Regression Analysis, Linear Models, and Related Methods*. Sage Publications, 1997.

[3] J Fox. Effect displays in r for generalised linear models. *Journal of Statistical Software*, 8(15):1–27, 2003.

[4] DC Montgomery. *Design and Analysis of Experiments*. Wiley, 2005.

[5] JA Nelder and WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, 135:370–384, 1972.

[6] JH Zar. *Biostatistical Analysis*. Prentice Hall, 4th edition, 1999.

[7] P Dixon. Models of accuracy in repeated-measures designs. *Journal of Memory and Language*, 59(4):447–456, 2008.

[8] F Harrell. *Regression Modeling Strategies*. Springer, 2001.

[9] D Hosmer and S Lemeshow. *Applied Logistic Regression*. New York: Wiley, 1989.

[10] EW Steyerberg. *Clinical Prediction Models*. Springer, 2009.